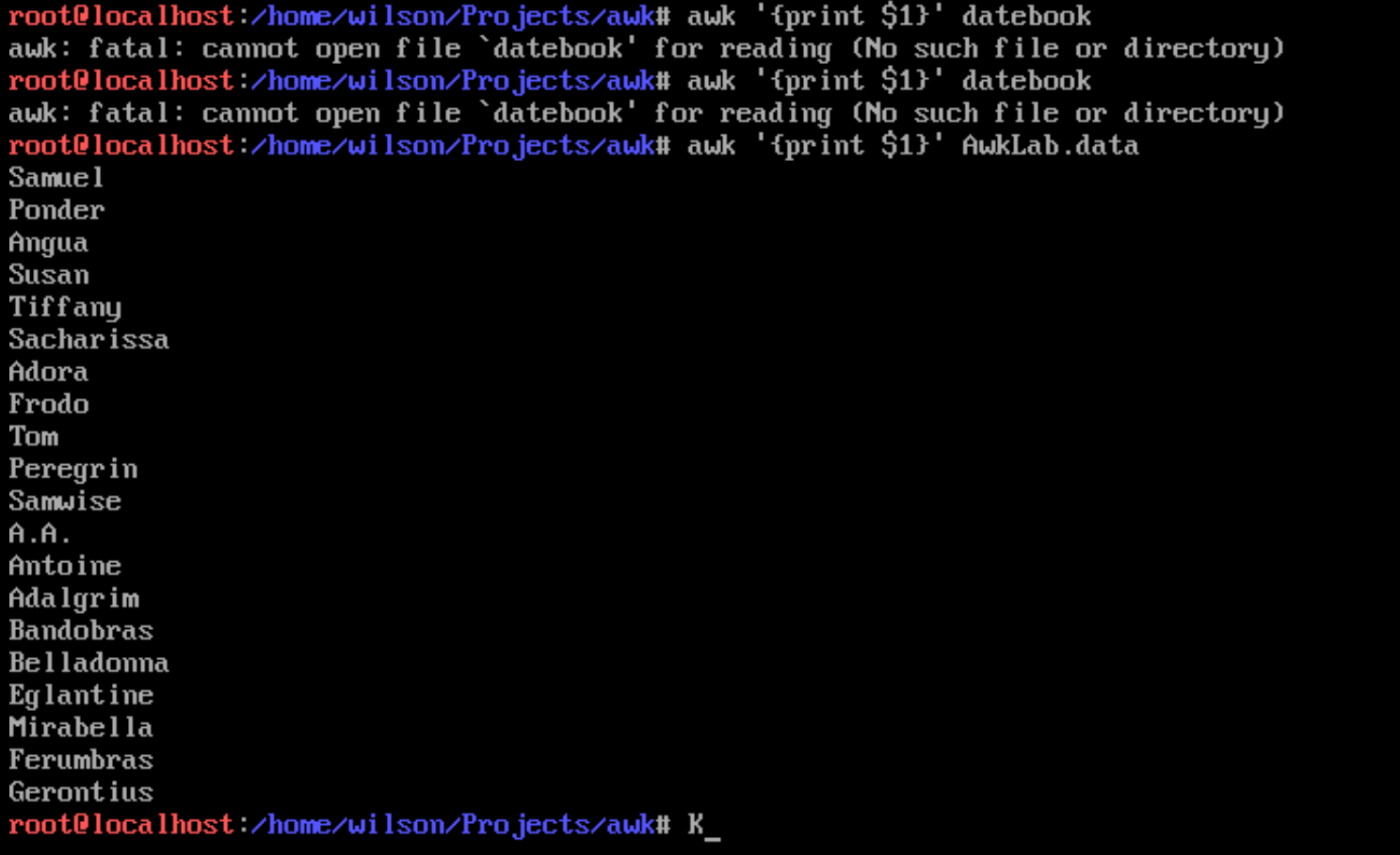
Awk assignment

1. Print all the First Names.

To print the first names I printed the first column using $ and enclosed this within curly brackets as a body block.

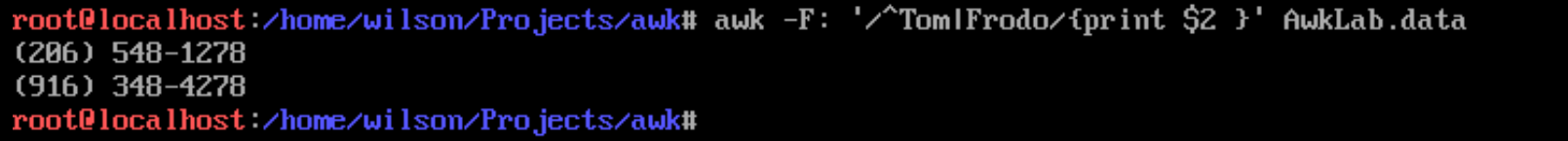


Commands used:

awk ‘{print $1}’ AwkLab.data

1. Print phone numbers for Tom and Frodo

To print the phone number for Tom and Frodo, I used awk with the -F parameter to declare separator. I then found lines which contain Tom or Frodo and then printed the second column (the phone numbers) of AwkLab.data.



Commands Used:

awk -F: ‘/^Tom|Frodo/{print $2 }’ AwkLab.data

1. Print Peregrin's name and phone number area code.

To print Peregrin’s name and phone number I used awk without parameters. I first found the line which starts with Peregrin and then printed the first and second column (Name and area code). I then added a space with double quotations to separate Peregrin’s first and last name with a space.

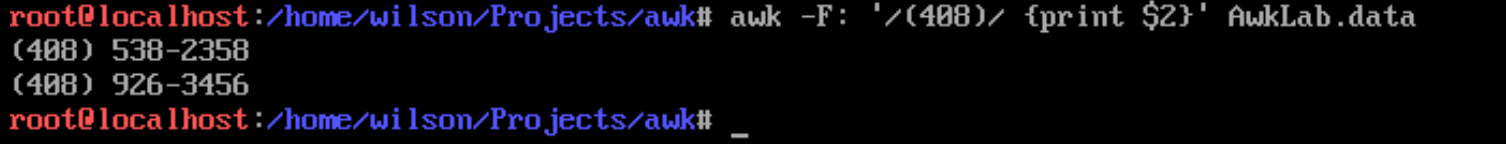


Commands used:

awk ‘/^Peregrin/{print $1 “ “ $2 }’ AwkLab.data

1. Print all phone numbers in the 408 area code.

To print the phone numbers with the 408 area code I first used awk with the -F parameter to declare a separator with a colon. I then printed the second column after the 408 area code.

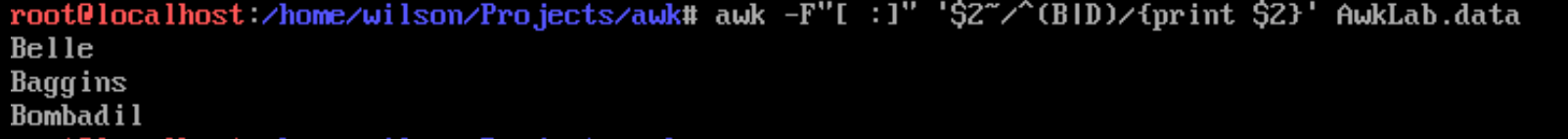


Commands used:

awk -F : ‘/(408)/ {print $2}’ AwkLab.data

1. Print all Last names beginning with either a B or D

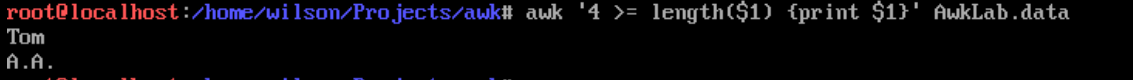
For this question I needed last names beginning with either a B or D. To do this I used F to create a separator that separates by either a space or colon. Now the second contains the last name so I used the tilde operator to match the last name with a regular expression starting with B or D. Then printed the second if it matched.



Commands used: awk -F"[ :]" '$2 ~ /^(B|D)/ {print $2}' Awklab.data

1. Print all first names containing four or less characters.

To print first names containing four or less characters I used the first column which contained first names and identified which names were four or less characters using the length function and comparison operators.

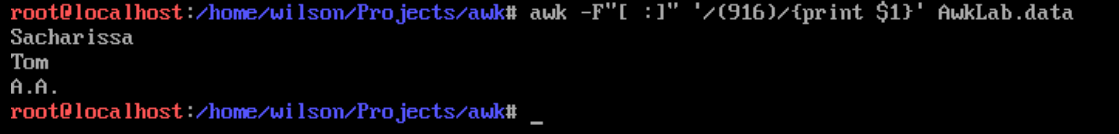


Commands used:

awk ‘4 >= length($1) {print $1}’ Awklab.data

1. Print the first names of all those in the 916 area code.

To find the first names in the 916 area code, I first used the -F to create a separator for spaces and colons. I then identified lines with 916 as area code and returned the first column which was the first name because of the separators.

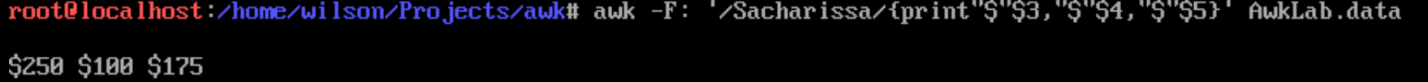


Commands Used:

awk -F”[ :]” ‘/(916)/{print $1}’ AwkLab.data

1. Print Sacharissa's campaign contributions. Each value should be printed with a leading dollar sign; e.g., $250 $100 $175.

First I made the colon separator then I identified the line with Sacharissa. I then printed a dollar sign and the column value of each campaign contribution number.

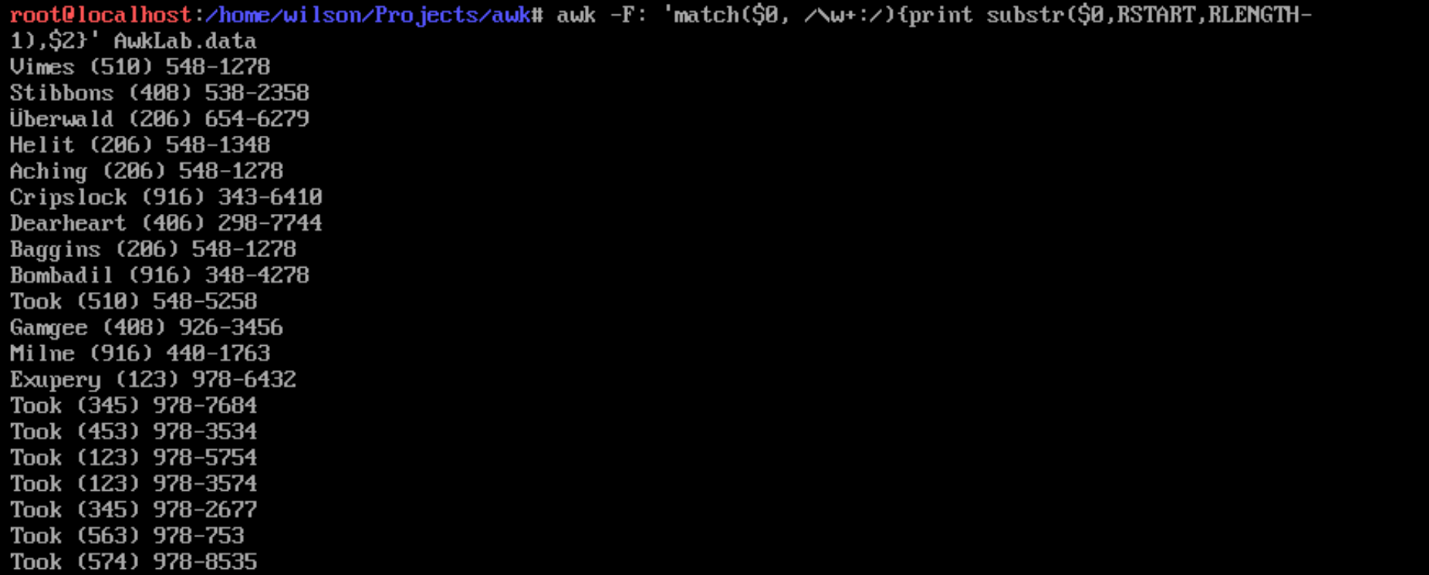


Commands used:

awk -F: ‘/Sacharissa/{print “$”$3,”$”$4,”$”$5}’ AwkLab.data

1. Print last names followed by a comma and the phone number. Be careful of the last names's format.

For this question I used awk with -F to create a separator using a colon. I then matched the entire line to the last name of the person. I used \w to identify any character as part of the last name. Then in the body I used the substring function to print out the match without the color and the phone number after.



Commands used:

awk -F: ‘match($0, /\w+:/){print substr($0,RSTART,RLENGTH-1),$1}’ AwkLab.data

1. Write an awk script called facts to do the following (MUST be an awk script not just a bash script or commands on the commandline)
2. Print the first and last names of those who contributed more than $110 in the last month.

For this question I needed to find the first and last names of those who contributed more than $110 in the last month. To do this I first used the awk command with -F and a separator using a colon. I then used $NF to identify the last field listed and see if it is more than 110. If it is more than 110, I print the first and last name of the person using the first column.

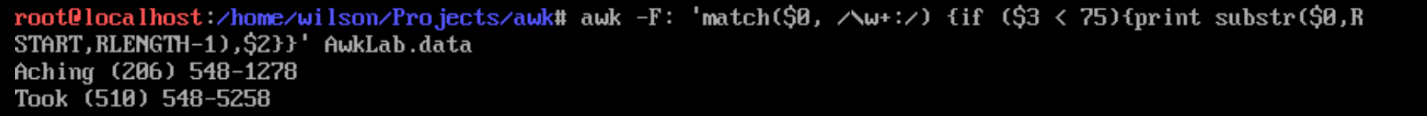


Commands used:

awk -F: ‘$NF > 110 {print $1}’ Awklab.data

1. Print the last names and phone numbers of those who contributed less than $75 in the first month.

To start, I first used awk with -F using a color separator. If the third column was less than 75, I matched the entire line to the last name of the person. I used \w to identify any character as part of the last name. Then in the body I used the substring function to print out the match without the color and the phone number after.



Commands used:

awk -F: ‘match($0, /\w+:/){if ($3 < 75){print substr($0,RSTART,RLENGTH-1),$1}}’ AwkLab.data

1. Print the first names of those who contributed between $75 and $150 in the first month.

For this command I used awk and -F separator using a colon and a space. I then found if the contributions in the third column subtracted by the final column (the first month) was more than 75 an less than 150. If it was between 75 and 150, it prints the first name of the person.

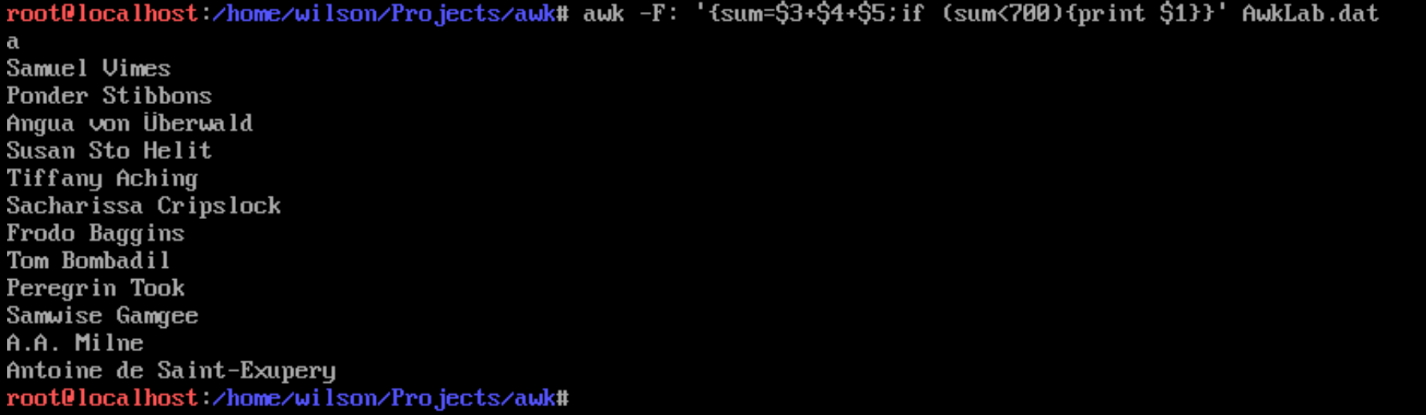


Commands used:

awk -F”[ :]” ‘$(NF-3) > 75 && $(NF-3) < 150 {print $(NF-3)}’ Awklab.data

1. Print the first and last names of those who contributed less than $700 over the three-month period.

For this question I needed to find all first and last names of people who contributed less than $700 over the three-month period. To do this I first used awk with an -F separator using a colon. I then created a variable of sum which combined all three months of contributions. I then used an if statement too find all people with a sum of less than 700. If the sum was less than 700, I printed the first column (first and last name)



Commands used:

awk -F: ‘{sum=$3+$4+$5; if (sum>700){print $1}} Awklab.data

1. Print the first names and first letter of the last name of those with an average monthly contribution greater than $100 .

I first started with awk -F to separate with a colon. I then used match to get the index of the first letter of the last name. Then in the body I create average variable combining the contribution columns and dividing by the amount of contributions. If any had an average greater than 100, I printed the name using the substring function from the start of the string to the first letter of the last name.

Commands used:

awk -F: ‘match($0, /\w+:/){avg=($3+$4+$5)/3; if (avg > 100){print substr($0,0,RSTART)}}’ AwkLab.data

1. Print the last name of those not in the 916 area code.

I first start with awk and the -F separator of colon. I then used a exclamation point to negate the 916 expression in the body. I used match to get the last name index and printed it using substring if 916 was not their area code.

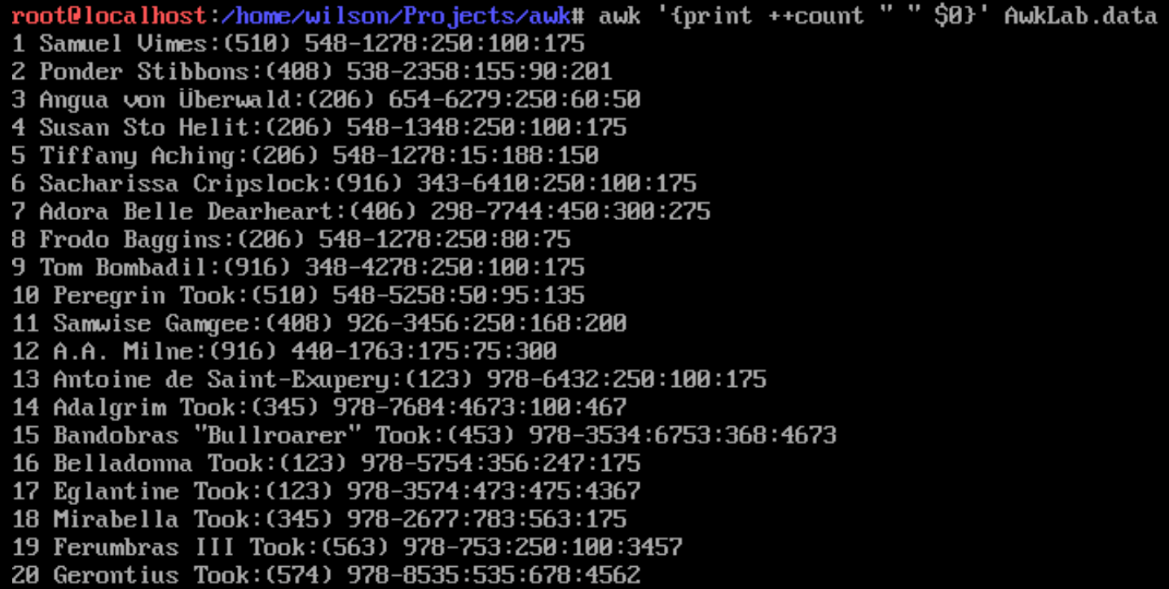


Commands used:

awk -F: ‘!/(916)/{match($0, /\w+:/);print substr($0,RSTART,RLENGTH-1)}’ AwkLab.data

1. Print each record preceded by the number of the record.

To print each record I created a count variable which incremented its value by one for every record. Then I printed the full record after.

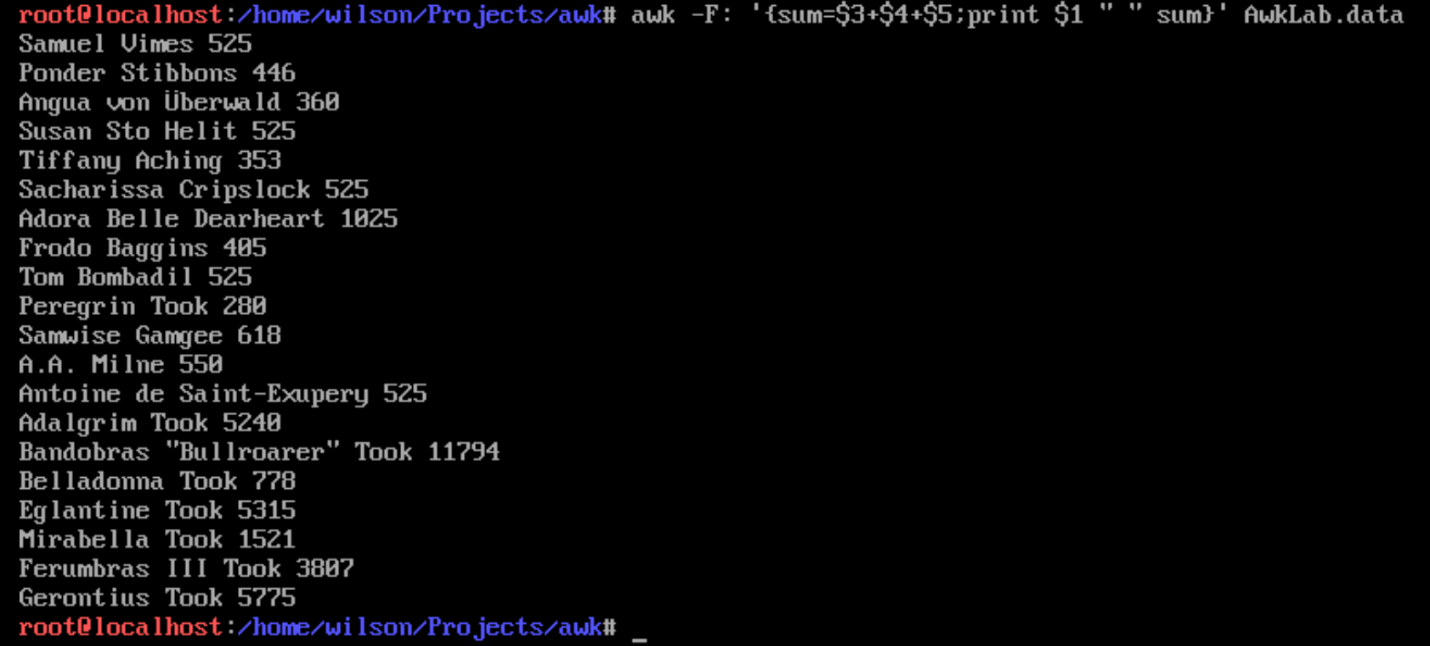


Commands used:

awk ‘{print ++count $0}’ AwkLab.data

1. Print the name and total contribution of each person.

First I used awk with a -F separator using a colon. I then identified sum as a variable combining the three contributions. After this I printed the first column(first and last name) and a space and then the sum of the contributions.



Commands used:

awk -F: ‘{sum= $3+$4+$5;print $1 “ “ sum}’ AwkLab.data

1. Add $10 to Tiffany Aching's first contribution.

To add $10 to Tiffany’s first contribution I first created a separator with a colon. I then found the lines starting with Tiffany and printed the third column(the first contribution) plus 10 to get $25

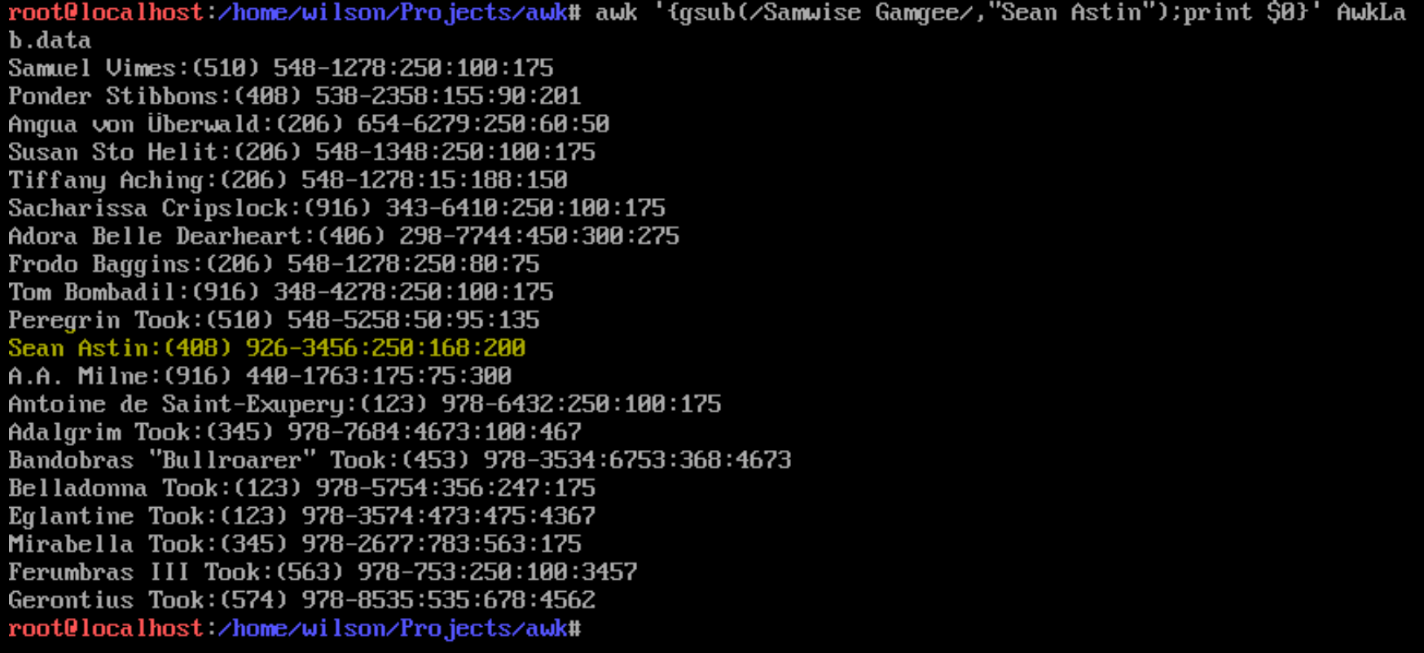


Commands used:

awk -F: ‘/Tiffany Aching/{print “$” $3+10}’ AwkLab.data

1. Change Samwise Gamgee's name to Sean Astin

For this question I used gsub to replace Samwise with Sean and then printed all lines.



Commands used:

awk ‘{gsub(/Samwise Gamgee/,”Sean Astin”)print}’ AwkLab.data

Sources used:

<https://man7.org/linux/man-pages/man1/awk.1p.html>

<https://flylib.com/books/en/4.356.1.52/1/>